Р.А. Бикташев, Е.И. Гурин // Известия вузов. Поволжский регион. Технические науки – 2007. – № 2. – С.3–12.

3. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э.Таненбаум, М. ван Стеен // СПБ: Питер, 2003. – 877 с.

4. Мартышкин А.И. Исследование алгоритмов планирования процессов в системах реального времени // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIII Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2015. – С. 118-124.

Мартышкин А.И., Мартенс-Атюшев Д.С. Разработка подсистемы планирования и назначения задач реконфигурируемой вычислительной системы для цифровой обработки сигнала // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIV Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2016. – С. 115-119.

УДК 004.451

АППАРАТНАЯ ПОДДЕРЖКА И ПРОВЕРКА ПРАВИЛЬНОСТИ РАБОТЫ ДИСПЕТЧЕРА ЗАДАЧ МНОГОПРОЦЕССОРНОЙ РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

А.И. Мартышкин
кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье представлена возможность аппаратной реализации и проверки правильности работы диспетчера задач многопроцессорной реконфигурируемой вычислительной системы Приводится методика верификации в программе SMV. В ходе исследования получена кодировка недетерминированного автомата НДА на входном языке SMV в соответствии с описанной методикой.

Ключевые слова: реконфигурируемая вычислительная система, диспетчер задач, верификация алгоритмов, синхронизация процессов, операционная система, программа SMV.

HARDWARE SUPPORT AND VERIFICATION OF THE OPERATION OF THE TASKMAN CONTROLLER OF A MULTI-PROCESSOR RECONFIGURABLE COMPUTING SYSTEM

A.I. Martyshkin
Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The article presents the possibility of hardware implementation and verification of the correctness of the work of the task manager of a multiprocessor reconfigurable

computing system. The verification method in the SMV program is given. In the course of the study, a non-deterministic NDA automaton encoding was obtained in the input language SMV in accordance with the described method.

Keywords: reconfigurable computing system, task manager, algorithm verification, process synchronization, operating system, SMV program.

Введение. При проектировании операционных систем для высокопроизводительных вычислительных (многопроцессорных и реконфигурируемых) систем, зачастую остро стоит проблема уменьшения накладных расходов, возникающих при планировании процессов. Частью планировщика является функция диспетчеризации задач (процессов, потоков) при их назначении по процессорным узлам. Реализация этой функции тесно связана с необходимостью синхронизации взаимодействующих процессов [1].

Цель работы. В статье приводится способ аппаратной поддержки и результаты проверки правильности функционирования диспетчера задач многопроцессорной реконфигурируемой вычислительной системы, в совокупности способствующие повышению быстродействия реконфигурируемой вычислительной системы.

Материал и результаты исследований. Примем, что t – время выполнения основной части программы в однопроцессорном режиме, а время выполнения этой же программы на N-процессорной реконфигурируемой системе составит

$$t^* = t/N. \quad (1)$$

Примем, что τ – время, необходимое для синхронизации процессов. Тогда время выполнения программы с учетом синхронизации для однопроцессорного варианта составит (t+τ), а для многопроцессорного соответственно (t*+ τ).

Коэффициент использования производительности η, показывающий какая доля мощности одного процессора задействована при реализации одного из параллельных процессов, в результате выполнения которого возникают временные потери при осуществлении синхронизации, составит

- для однопроцессорного варианта

$$\eta = t/(t+\tau), \quad (2)$$

- для N – процессорного варианта

$$\eta^* = t^*/(t^*+\tau) = t/(t+N\tau), \quad (3)$$

где τ и Nτ – непроизводительное время, затрачиваемое на синхронизацию процессов в однопроцессорном и многопроцессорном режимах работы. Как следует из (2) и (3) временные затраты на синхронизацию в многопроцессорном режиме увеличиваются относительно однопроцессорного в N раз.

Традиционно планировщик и диспетчер задач выполняются программно и реализуются в SMP-системах методом вызова этих функций из общей памяти, где хранится программа операционной системы [1]. Практическая программная реализация предполагает два варианта: 1) в пространстве пользователя; 2) в пространстве ядра.

Программная реализация в пространстве пользователя хотя и является быстрой, но вызывает некоторые затруднения, ввиду того, что для выполнения сопряженной с диспетчеризацией процедуры синхронизации необходимы 3 семафора: один – счётчик для подсчета числа мест, занятых готовыми для выполнения задачами; другой – счётчик для подсчета числа процессоров, занятых обслуживанием; третий – мьютекс для реализации взаимного исключения, предотвращающего одновременный доступ нескольких свободных процессоров к единственной (глобальной) очереди, в данном случае играющей роль общего ресурса. Очевидным и безусловным выходом из такой ситуации является реализация процедуры синхронизации в пространстве ядра. Однако возрастающие при этом временные затраты резко уменьшают производительность реконфигурируемой системы.

В статье для проверки алгоритма работы диспетчера задач предлагается использовать систему SMV (Symbolic Model Verifier) [2] фирмы Cadence, которая широко применяется в промышленности и научных исследованиях. В частности, такие известные фирмы, как AT&T, Fujitsu, Intel, IBM, Motorola и другие используют SMV для верификации алгоритмов функционирования своих разработок. В предлагаемой методике верификации недетерминированный автомат (НДА) [3] описывается в соответствии с определенными правилами на входном языке SMV, а проверяемые свойства системы специфицируются в виде формул временных логик CTL в терминах состояний НДА. Важной задачей при этом является разработка адекватного описания НДА на входном языке SMV. Возможность подобного подхода подтверждается существованием методик верификации в SMV для других моделей переходов состояний, например, цветных сетей Петри [4].

Ниже приведена методика верификации систем, построенных на основе НДА, с использованием метода Model Checking в системе SMV. Основные шаги методики представлены на рисунке 1, особенность которой состоит в том, что она (в отличие, например, от [5]) не требуется явного построения структуры Крипке для модели НДА, включающей все возможные состояния, поскольку система SMV сама «развертывает» структуру Крипке (в виде BDD) в процессе доказательства исходя из символьного описания на входном языке. Возможность подобного подхода подтверждается существованием методик верификации в SMV для моделей переходов состояний, отличных от НДА, например, цветных сетей Петри [4]. Тем не менее,

возможен подход к верификации систем на основе НДА, в котором используется «явная» структура Крипке, в качестве которой выступает НДА после проведения процедуры детерминизации (то есть детерминированный автомат) [6, 7]. Важным моментом в предложенной методике является построение адекватного описания НДА на входном языке SMV. Основой такой «адекватности» является то, что как язык SMV (в рамках процесса), так и НДА обладают свойствами синхронных моделей.

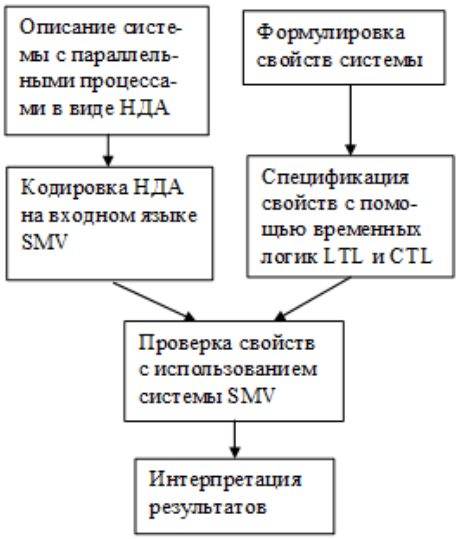

Рисунок 1 – Методика верификации систем на основе НДА

Правила описания СКУ НДА на языке SVM просты. НДА оформляется в виде одного модуля SMV. Входные переменные, а также события НДА, зависящие от внешнего окружения, представляются как параметры модуля. Считается, что они могут принимать произвольные значения. Каждому событию в НДА ставится в соответствие булева переменная. Начальные единичные значения устанавливаются для тех переменных, которые соответствуют начальным состояниям. Начальные значения остальных булевых переменных устанавливаются в ноль.

Реализации режимов A и B несколько различаются. Сначала рассмотрим основной режим (режим A). Уравнения СКУ один к одному транслируются в операторы присваивания SMV, причем для представления события, стоящего в левой части уравнения, используется конструкция next в левой части оператора присваивания. Правая часть уравнения СКУ определяется как логическое выражение языка SMV. Операторы присваивания, соответствующие СКУ, объединяются в составной оператор, который предваряется конструкцией default. Операторы присваивания, стоящие в этой конструкции, сбрасывают все переменные, соответствующие событиям НДА, в ноль. Смысл default-оператора в данном контексте означает, что все события, которые не устанавливаются в следующем такте работы системы, удаляются

из совокупного состояния. Для представления НДА, функционирующего в режиме B, для каждого события si НДА вводятся два условия: условие установки (set_si) и условие сброса (reset_si) события. Условия установки события эквиваленты правой части соответствующего уравнения СКУ. Условия сброса определяются поиском сбрасываемого события в правых частях в системе уравнений СКУ и определения условий установки событий, стоящих в левых частях найденных уравнений. Условие сброса формируется путем дизъюнкции этих условий установки. Для формирования условий сброса удобно пользоваться сетевым представлением НДА, позволяющим наглядно представить причинно-следственные связи между событиями. Для каждого события si формируется оператор присваивания SMV следующего вида: next(si):= set_si | (~set_si & ~reset_si & si), вычисляющий статус события si в следующий момент времени.

Свойства системы первоначально формулируются на естественном языке. Все свойства можно разделить на общие и частные. В отношении систем с параллельными процессами обычно выделяют следующие общие свойства: безопасности, живости и справедливости (fairness). В системе SMV свойства формулируются с помощью линейной временной логики LTL или логики дерева вычислений CTL [8]. В логике LTL определены линейные временные операторы X (в следующий момент времени), F (возможно в будущем), G (всегда) и U (пока не). В логике CTL в дополнение к этому используется квантификация путей. При этом кванторы E и A предваряют линейные временные операторы (например, AG, EF, AX и т.п.). Атомарные высказывания формулируются на основе событий НДА. Причем существует два элементарных вида высказываний: событие НДА входит в совокупное состояние и событие не входит в совокупное состояние.

Перейдём непосредственно к верификации алгоритма синхронизации процессов при диспетчеризации задач в многопроцессорных реконфигурируемых системах с использованием механизма рандеву [3]. Система канонических уравнений, описывающих данный алгоритм, включает в себя три части: процесс «клиент», процесс «сервер», и система канонических уравнений, описывающих события после рандеву.

Ниже представлен фрагмент кодировки НДА (режим A) на входном языке SMV, полученный в соответствии с методикой, описанной ранее в этом разделе. Количество процессоров в системе возьмём равным 4.

```
next(Sp_0):= Sp1_0 | Sp2_0 | Sp3_0 | Sp4_0;
next(Sp1_BK):=Sp1_0 & Sk_0 & Sp1_P;
next(Sp2_BK):=Sp2_0 & Sk_0 & Sp2_P;
next(Sp3_BK):=Sp3_0 & Sk_0 & Sp3_P;
next(Sp4_BK):=Sp4_0 & Sk_0 & Sp4_P;
next(Sp1_S):=(Sp1_0 & (~Sk_0)) | (Sp1_S & Sp1_P & (~Sk_BP1));
```

```
next(Sp2_S):=(Sp2_0 & (~Sk_0)) | (Sp2_S & Sp2_P & (~Sk_BP2));
next(Sp3_S):=(Sp3_0 & (~Sk_0)) | (Sp3_S & Sp3_P & (~Sk_BP3));
next(Sp4_S):=(Sp4_0 & (~Sk_0)) | (Sp4_S & Sp4_P & (~Sk_BP4));
next(Sp1_OK):=(Sp1_S) & Sp1_P & Sk_BP1;
next(Sp2_OK):=(Sp2_S) & Sp2_P & Sk_BP2;
next(Sp3_OK):=(Sp3_S) & Sp3_P & Sk_BP3;
next(Sp4_OK):=(Sp4_S) & Sp4_P & Sk_BP4;
next(Sp1_g):=((Sp1_BK | Sp1_OK) & Sk_KP1) | (Sp1_g & (~Sk_g));
next(Sp2_g):=((Sp2_BK | Sp2_OK) & Sk_KP2) | (Sp2_g & (~Sk_g));
next(Sp3_g):=((Sp3_BK | Sp3_OK) & Sk_KP3) | (Sp3_g & (~Sk_g));
next(Sp4_g):=((Sp4_BK | Sp4_OK) & Sk_KP4) | (Sp4_g & (~Sk_g));
```

Проверим свойство достижимости состояний в процессе «клиент», процессе «сервер» (для одного процессора) и процессе «после рандеву». Данные свойства выражаются с помощью следующих формул временной логики:

```
SPEC EF Sk_0;
SPEC EF Sk_BP1;
SPEC EF Sk_KP1;
SPEC EF Sk_g;
SPEC EF Sk_S;
SPEC EF Sp_0;
SPEC EF Sp1_BK;
SPEC EF Sp1_S;
SPEC EF Sp1_OK;
SPEC EF Sp1_g;
SPEC EF Sp1_ST;
SPEC EF Sk_OK;
SPEC EF Sp1_OD;
SPEC EF Sp1_ZD;
SPEC EF Sk_U;
```

SMV дает следующий ответ (рисунок 2): все указанные состояния достижимы при определенных начальных условиях, то есть, в системе нет состояний, которые никогда бы не были достижимы из начальных состояний.

Вывод. Следуя вышесказанному, сделаем вывод, что аппаратная реализация алгоритмов планирования дает возможность быстрее определять задачу, которая должна получить процессорное время следующей, чем программная реализация тех же самых алгоритмов внутри планировщика операционной системы.

Работа выполнена при финансовой поддержке стипендии Президента РФ молодым ученым и аспирантам на 2018-2020 гг. (СП-68.2018.5).

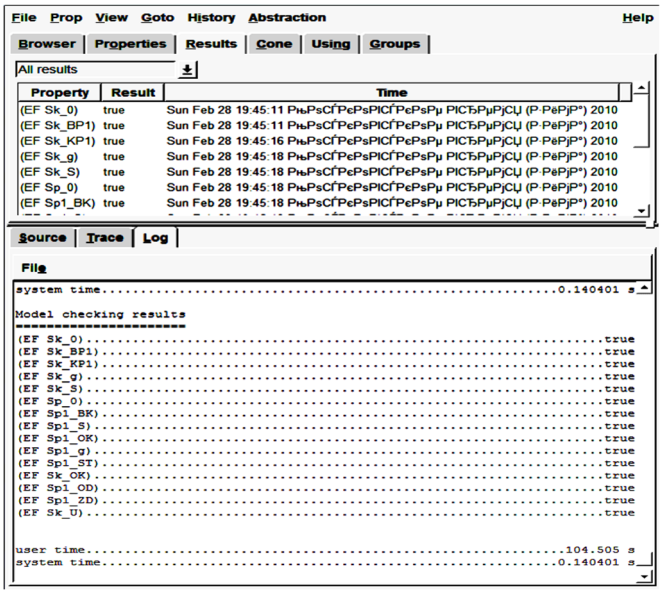

Рисунок 2 – Верификация НДА алгоритма функционирования диспетчера в системе SMV

ЛИТЕРАТУРА

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. – СПб.: Питер, 2015. – 1120 с.: ил. – (Серия «Классика computer science»).

2. SMV. Symbolic Model Verifier. [Электронный ресурс]. URL: www.cs.cmu.edu (дата обращения: 21.03.2019).

3. Вашкевич, Н. П. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации: монография / Н. П. Вашкевич, Р. А. Бикташев. – Пенза: Изд-во ПГУ, 2016. – 394 с.

4. Stotts D., Navon J. Model checking cobweb protocols for verification of HTML frames behavior // Proc. 11th Int.Conf. on World Wide Web (WWW'02), Honolulu, Hawaii, 2002. - P.182-190

5. Кузьмин Е.В., Соколов В.А. Моделирование, спецификация и верификация "автоматных" программ // Программирование. – № 1. – 2008. – С.38-60.

6. Вашкевич Н.П. Недетерминированные автоматы в проектировании систем параллельной обработки: учеб. пособие. - Пенза: изд-во Пенз. гос.ун-та, 2004. – 280 с.

7. Вашкевич Н.П. Недетерминированные автоматы и их использование для синтеза систем управления. Ч.1 / Н.П. Вашкевич, С.Н. Вашкевич // Эквивалентные преобразования недетерминированных автоматов: Учеб. пособие. – Пенза: Изд-во Пенз. гос. ун-та, 1996. – 88 с.

8. Кларк Э., Грамберг О., Пелед Д. Верификация моделей программ: Model Checking. М.: МЦНМО. 2002.